2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

Extensive Analysis of Linear Complementarity Problem (LCP) Solver Performance on Randomly Generated Rigid Body Contact Problems

Evan Drumwright and Dylan A. Shell

Abstract— **The Linear Complementarity Problem (LCP) is a key problem in robot dynamics, optimization, and simulation. Common experience with dynamic robotic simulations suggests that the numerical robustness of the LCP solver often determines simulation usability: if the solver fails to find a solution or finds a solution with significant residual error, interpenetration can result, the simulation can gain energy, or both. This paper undertakes the first comprehensive evaluation of LCP solvers across the space of multi-rigid body contact problems. We evaluate the performance of these solvers along the dimensions of solubility, solution quality, and running time.**

I. INTRODUCTION

The Linear Complementarity Problem (LCP) is omnipresent in rigid body contact problems. Simulation robustness and performance are directly affected by the particular solver employed: important considerations include running-time, solution quality, and reliability. This work seeks a systematic, comparative understanding of the available solvers *applied to contact problems*.

Dynamic robotic simulation has the potential to be one of the roboticist's greatest tools. As a via point between purely kinematic simulations and physically situated robotic experiments, dynamic simulation fills important roles in testing and debugging robot code and providing substrates for robot learning. However, effective dynamic robotic simulation has remained largely limited to applications with limited or no contact. Though reports of simulating robots with contact exist in the literature, practical difficulties abound before such simulation becomes simple, robust, and ubiquitous.

One of the greatest obstacles toward practical dynamic robotic simulation with contact lies with the nonlinear optimization solvers used to solve multi-rigid body contact problems. Though the contact models to which such algorithms are applied are solvable in theory, experience indicates the solvers are prone to failure. These failures can lead to interpenetrating rigid bodies (from which recovery is difficult or impossible), simulations becoming unstable, and sliding (rather than sticking) contacts. The practical effects of these failures are robots that "sink" into the ground, simulations that crash, and objects that slip out of grippers.

The failures of nonlinear optimization codes on multi-rigid body simulation contact problems can be placed into two categories: failures to produce any solution and failures to produce a solution with residual error below a desired tolerance. Both kinds of failures can lead to the deleterious phenomena discussed above, though the latter failures may be less critical on simulations running at shorter timescales. That failures occur raises a number of questions: Can the factors that lead to failure be identified and mitigated? Are some algorithms for solving these contact problems more effective than others? Are some contact models (*i.e.*, convex *vs*. non-convex models) more amenable to solution?

Evan Drumwright is with the Department of Computer Science, The George Washington University. drum@gwu.edu

Dylan A. Shell is with the Department of Computer Science and Engineering, Texas A&M University. dshell@cse.tamu.edu

II. BACKGROUND

LCPs have been used to model rigid body contact scenarios since the works of Löstedt [1], Moreau [2], and Baraff [3]. The most influential work in this area (measured by presence in current rigid body simulation libraries) is that of Stewart and Trinkle [4], [5] and Anitescu and Potra [6], [7].

State of the art methods for modeling multiple simultaneous contacts in robotic simulation, including the works of Stewart and Trinkle [4], [5], Anitescu and Potra [6], [7], and Drumwright and Shell [8], [9] use nonlinear programming to solve for unknown contact impulses. The solution methods differ considerably, however: solving models from the first two groups requires special algorithms (methods for LCPs with *copositive* matrices[1]). Any algorithm capable of solving *monotone* LCPs (equivalent to convex quadratic programs) is suitable for the methods of Drumwright and Shell.[2]

III. SOLVING LCPs

The particular contact model employed dictates the set of feasible LCP algorithms. We describe some algorithms for solving LCPs in the next section.

A. Linear complementarity problem solvers

By the concept of Lagrangian duality derived from the KKT conditions, every quadratic program (QP) is equivalent to an LCP: methods developed toward solving either problem are applicable to both. We describe some prominent algorithms for solving LCPs (and QPs by extension) below:

1) Pivoting solvers: A number of pivoting solvers, including (but not limited to) those of Dantzig and Cottle [10], [11], Murty [12], and Lemke [13], operate under the pivoting principle introduced by the Simplex algorithm. Every pivoting algorithm solves a particular class of LCP:

[1] A real matrix is copositive if $\boldsymbol{x}^{\mathrm{T}}\mathbf{A}\boldsymbol{x} \geq 0, \forall \boldsymbol{x} \geq 0$. The set of copositive matrices includes the set of positive-definite matrices.

[2] In this work, we assume that all contact models employ linearized friction cones to facilitate comparison: if circular or elliptical friction cones are used instead, Drumwright and Shell [9] require an algorithm for solving convex quadratically constrained quadratic programs while Stewart and Trinkle [5] require an algorithm for solving nonlinear complementarity problems.

978-1-4673-1736-8/12/S31.00 ©2012 IEEE

Dantzig and Cottle's method solves LCPs with $\mathbf{P}$-matrices (matrices with positive principal minors) and positive semi-definite matrices, Murty's least-index method solves problems with $\mathbf{P}$-matrices, and Lemke's algorithm solves LCPs with semimonotone matrices (a large class of matrices that includes copositive matrices, matrices with non-negative diagonals, and matrices with non-negative principal minors). One disadvantage of pivoting schemes is that an exponential number of pivoting operations may be required (LCPs can be constructed to exhibit this behavior for any possible pivoting rule), though n pivoting operations are expected for an order n LCP with the algorithms above [14]. We use Lemke's algorithm in our experiments with pivoting algorithms because it is widely available in various implementations and because it is capable of solving a large class of problems.

2) Interior-point solvers: Interior-point solvers, also known as *barrier methods* [15], combine Newton's method for unconstrained optimization with a logarithmic barrier in the objective function to prevent constraint violation. Interior-point methods are often faster for QPs with many variables and constraints than active-set methods (algorithms that track the subset of the m inequality constraints $g_i(x) \geq 0, i \in 1 \ldots m$ for which $g_i(x) = 0$) for QPs and pivoting methods for LCPs [16]. Interior-point methods exhibit worst-case polynomial complexity on convex nonlinear optimization problems [15].

3) PATH: PATH [17] is a free, closed source, commercial quality solver for mixed complementarity problems (*i.e.*, combined equality constraints with both LCPs and nonlinear complementarity problems). PATH employs a stabilized Newton method for solving these complementarity problems via a formulation as a set of nonlinear equations. The use of PATH for solving multi-rigid body contact problems is frequently reported in the literature (*e.g.*, [18], [19], [20]).

4) Iterative solvers: Iterative methods are most advantageous for large, sparse LCPs and QPs. We initially experimented with *projected successive overrelaxation* (SOR) splitting method [21], [14] for solving monotone and copositive LCPs. Although convergence for iterative methods is guaranteed only for LCPs with symmetric, strictly copositive matrices [21], such approaches are popular for solving "harder" LCPs with bisymmetric copositive matrices (*e.g.*, the rigid body simulation library ODE uses this approach in its "quick step" method) due to potential for rapid convergence. Lacoursiére's extensive numerical experiments [22] on splitting methods for frictional contact problems indicate the difficulty of employing such approaches in this application. Performance of these approaches on our randomly generated contact problems was so poor (solution rates below 50% by our criteria and high error residuals) that we cannot recommend employing such solvers in robotics applications.

B. Regularization

Though a LCP may be solvable in theory by one or more of the algorithms above, an implementation may fail to solve it in practice due to numerical issues. In such cases, a *regularization* strategy that trades some solution error for solubility is advisable. Cottle [14] describes a strategy for regularization that is identical to the Tikhonov regularization [23] method for least squares problems. This approach adds a small diagonal matrix (typically ϵI) to the LCP matrix.

Algorithm III.1 Generation of monotone LCPs corresponding to random contact problems

Input:

The *class* of generalized inertia matrix, either *reduced-rank* or *full-rank*.

Output:

An LCP($\boldsymbol{q}, \mathbf{M}$) where $\mathbf{M}$ is positive semi-definite.

1: Select # of generalized coordinates (n_{gc}) uniformly from integers $[6, 15]$
2: Select # of contact points (n_c) uniformly from integers $[2, 20]$
3: Select # of redundant contacts (n_{rc}) uniformly from integers $[0, n_c - 1]$
4: Generate n_{gc}-dimensional vector ($\boldsymbol{v}$) with each component $v_i \sim \mathcal{U}(-1, 1)$.
5: Generate $n_{gc} \times n_c$ matrix ($\mathbf{N}$) with each component $n_{ij} \sim \mathcal{U}(-1, 1)$.
6: Randomly introduce n_{rc} linearly dependent columns into $\mathbf{N}$ using integral combinations of the $(n_c - n_{rc})$ linearly independent columns.
7: Generate $n_{gc} \times n_{gc}$ symmetric generalized inertia matrix ($\mathbf{G}$), depending on class:
8: **if** *class* = *full-rank* **then**
9: Set $\mathbf{G}$ to the positive definite matrix generated though summing n_{gc} rank-1 updates.
10: **else if** *class* = *reduced-rank* **then**
11: Select rank of the generalized inertia matrix (k) uniformly from integers $[6, n_{gc}]$
12: Set $\mathbf{G}$ to the positive semi-definite matrix generated though summing k rank-1 updates.
13: Return the LCP($\mathbf{N}^\mathsf{T}\boldsymbol{v}, \mathbf{N}^\mathsf{T}\mathbf{G}^\star\mathbf{N}$).

Definitions:

- A rank-1 update to $n \times n$ matrix $\mathbf{A}$ is $\mathbf{A} \leftarrow \alpha \boldsymbol{x}\boldsymbol{y}^\mathsf{T} + \mathbf{A}$.
- Here $\mathbf{A}^\star$ denotes a SVD-regularized inverse of $\mathbf{A}$.

IV. EXPERIMENTS

The experiments that we describe in this section were based on the observation that we can create random contact problems even though we are presently unable to provide a corresponding physical scenario (kinematics of the mechanism or robot, geometric description of the bodies in contact): in general, there are no constraints on the contact Jacobians of a contact problem (*i.e.*, no special matrix structure), nor are there differential constraints on the generalized velocities of the bodies in a rigid body contact problem. Indeed, the only constraints present in a multi-rigid body contact problem are the symmetry and positive-definiteness of the generalized inertia matrix and the non-negativity of the coefficients of friction. Otherwise, both the dimensionality and range of values of the matrices and vectors in a contact problem can be determined randomly.

20,000 trials were conducted for each of the experiments listed below. Future work will attempt to assess the margin of error of our sample size (uncountable population size precludes the use of standard tools from sampling theory).

Algorithm IV.1 Generation of copositive LCPs corresponding to random contact problems

Input:

The *class* of generalized inertia matrix, one of either *reduced-rank* or *full-rank*.

Output:

An LCP($\boldsymbol{q}, \mathbf{M}$) where $\mathbf{M}$ is copositive; that is for all non-negative vectors $\boldsymbol{v}$, $\boldsymbol{v}^\mathsf{T}\mathbf{M}\boldsymbol{v} \geq 0$.

1: Select # of generalized coordinates (n_c) uniformly from integers $[6, 15]$
2: Select # of contact points (n_{gc}) uniformly from integers $[2, 20]$
3: Select # of polygon edges in the friction cone (n_k) uniformly from even integers $[4, 24]$
4: Generate n_c-dimensional vector ($\boldsymbol{\mu}$) with each component $\mu_i \sim \mathcal{U}(0, 1)$.
5: Generate n_{gc}-dimensional vector ($\boldsymbol{v}$) with each component $v_i \sim \mathcal{U}(-1, 1)$.
6: Generate $n_{gc} \times n_c$ matrix ($\mathbf{N}$) with each component $n_{ij} \sim \mathcal{U}(-1, 1)$.
7: Generate $n_{gc} \times (n_c n_k)$ matrix ($\mathbf{D}$) with each component $d_{ij} \sim \mathcal{U}(-1, 1)$. The matrix $\mathbf{D}$ is constructed such that, for every column $\boldsymbol{d}$, the column $-\boldsymbol{d}$ also appears in $\mathbf{D}$.
8: Randomly make some columns of $\mathbf{N}$ (and corresponding columns of $\mathbf{D}$) linearly dependent.
9: Generate $n_{gc} \times n_{gc}$ symmetric generalized inertia matrix ($\mathbf{G}$), depending on class:
10: **if** *class* = *full-rank* **then**
11: Set $\mathbf{G}$ to the positive definite matrix generated though summing n_{gc} rank-1 updates.
12: **else if** *class* = *reduced-rank* **then**
13: Set k to the rank of the generalized inertia matrix. Draw k uniformly from integers $[6, n_{gc}]$
14: Set $\mathbf{G}$ to the positive semi-definite matrix generated though summing k rank-1 updates.
15: Return the LCP as in Anitescu-Potra [6].

A. Evaluation criteria

We evaluate each solver with respect to three criteria: solubility, running time, and normal constraint violation (*i.e.*, the velocity that the bodies are moving toward one another in the normal direction using a given solution). Solubility is the most difficult of the three criteria to define strictly; for example, a solver may report success on a given contact problem, yet the solution it provides would lead to significant interpenetration. We mark a contact problem as solved if the change in work and constraint violation that would result from application of the determined forces are less than $1e^{-3}J$ and $1e^{-3}m/s$, respectively. Thus, if either a significant gain in simulation kinetic energy or significant movement toward interpenetration would result, we classify a "solution" as a failure. Although these are arbitrary thresholds, they do permit comparisons of relative performance between the examined methods.

B. Solver implementation details and parameterization

Numerical algorithms invariably use one or more tolerance parameters. We searched extensively over the usable range of parameters (described below) for each solver to evaluate its performance using its best parameterization as well as discern the importance of parameter selection for a solver.

Monotone LCP Solution Frequency					
			Parameterization		
Ex.	Method		Worst	Average	Best
1	Lemke		92.71%	97.59%	100.0%
1	PATH		97.48%	98.68%	99.27%
1	IP		99.90%	100.0%	100.0%
2	Lemke		79.73%	93.13%	100.0%
2	PATH		93.92%	94.80%	97.28%
2	IP		96.74%	96.75%	96.77%
3	Lemke		62.90%	63.57%	64.09%
3	PATH		61.81%	63.47%	63.93%
3	IP		63.70%	63.71%	63.72%
4	Lemke		55.83%	67.55%	73.25%
4	PATH		67.91%	68.95%	72.56%
4	IP		65.49%	65.61%	65.67%
5	Lemke	$[1, 10^5]$	69.89%	88.91%	100.0%
5		$[1, 10^8]$	59.35%	84.74%	99.99%
5		$[1, 10^{10}]$	54.05%	81.57%	99.94%
5	PATH	$[1, 10^5]$	94.07%	95.30%	99.20%
5		$[1, 10^8]$	94.03%	95.51%	99.19%
5		$[1, 10^{10}]$	91.72%	92.99%	96.99%
5	IP	$[1, 10^5]$	87.34%	87.54%	87.65%
5		$[1, 10^8]$	74.86%	75.05%	75.15%
5		$[1, 10^{10}]$	70.41%	70.61%	70.74%
6	Lemke	$[1, 10^5]$	94.13%	98.87%	100.0%
6		$[1, 10^8]$	78.85%	95.93%	100.0%
6		$[1, 10^{10}]$	71.10%	92.87%	99.99%
6	PATH	$[1, 10^5]$	98.82%	99.79%	99.92%
6		$[1, 10^8]$	98.58%	99.66%	99.87%
6		$[1, 10^{10}]$	96.81%	97.94%	98.17%
6	IP	$[1, 10^5]$	88.26%	88.35%	88.41%
6		$[1, 10^8]$	75.69%	75.91%	75.97%
6		$[1, 10^{10}]$	71.30%	71.49%	71.60%

TABLE IV.1

SUMMARY OF SOLVER PERFORMANCE ACROSS SETS OF MONOTONE LCPS FOR THE SIX EXPERIMENTS. BEST AND WORST COLUMNS GIVE SOLUTION FREQUENCY FOR THE PARAMETERIZATION WITH THE GREATEST AND LEAST SUCCESS, RESPECTIVELY. AVERAGE SOLUTION FREQUENCY IS OVER TOTAL RANGE OF EXAMINED PARAMETERS.

The maximum number of iterations for each solver is configurable. We limited every solver to $\max(1000, n^2)$ iterations where n is the order of the LCP.

1) Lemke's algorithm: Our implementation of Lemke's algorithm is a direct translation of the LEMKE code [24] from Matlab to C++. This code uses a "pivoting" tolerance (used to identify potential pivots) and a "zero" tolerance (used to determine minimal ratios for pivoting). Parameter selection ranged from $1e^{-20}$ to $1e^{-4}$ with steps of $1e^{+2}$.

2) Interior-point method: The interior-point method employed in the experiments is our implementation of the primal-dual method described in [15] (*i.e.*, we applied a standard convex programming approach to the Lagrangian dual of the LCP rather than using, *e.g.*, the method described by Cottle [14]). More sophisticated implementations could yield lower running time and higher solubility. Our implementation uses two parameters, ϵ and ϵ_{feas}, which represent the general solution tolerance and the solution tolerance for feasibility. Parameter selection range from $1e^{-20}$ to $1e^{-4}$ with steps of $1e^{+6}$ (parameter search was more coarse than that of the other solver implementations due to the considerably longer running times of the interior-point solver).

3) PATH: The PATH solver utilizes a single "convergence" parameter, which ranged from $1e^{-20}$ to $1e^{-4}$ using an iteration step of $1e^{+2}$ in our experiments .

C. Experiment 1: establishing a baseline

We used full-rank generalized inertia matrices and contact normal and tangent Jacobian matrices (*i.e.*, $\mathbf{N}$ and $\mathbf{D}$) with full row rank to establish baseline LCP solver performance on the randomly generated contact problems.

An $n \times n$ inertia matrix is generated using a summation of n randomly generated rank-1 updates to the zero matrix. The contact normal and tangent Jacobian matrices are filled with values randomly selected from the interval $[-1, 1]$ (with the provision that, for every column vector $\boldsymbol{d}$ in the tangent Jacobian, the vector $-\boldsymbol{d}$ is also present in that Jacobian).

Note that when there are more contact points than generalized coordinates, the matrix $\mathbf{N}^\mathsf{T}\mathbf{G}^{-1}\mathbf{N}$—which is used in both the monotone and copositive contact problems—will be rank-deficient even when both $\mathbf{N}$ and $\mathbf{G}$ are of full rank.

D. Experiment 2: full rank generalized inertia matrices, reduced row rank contact Jacobians

The solution existence guarantee for the Anitescu-Potra contact model depends on the contact normal and tangenti Jacobian matrices (*i.e.*, $\mathbf{N}$ and $\mathbf{D}$) having full row rank [6]. Our initial hypothesis was that failure of these matrices to have full row rank—resulting from the common existence of "redundant" contacts—might be responsible for most solver failures. This hypothesis follows from the observation that the matrix $\mathbf{N}^\mathsf{T}\mathbf{G}^{-1}\mathbf{N}$, which is employed in both algorithms, is singular when $\mathbf{N}$ does not have full row rank.

A reduced row rank contact Jacobian was produced by making the randomly-selected *dependent* rows an integral combination of the *independent* rows. The same independent and dependent row indices were used to compose both the normal and tangent contact Jacobians for a given trial.

E. Experiment 3: reduced rank generalized inertia matrices, full row rank contact Jacobians

This experiment tested the hypothesis that the generalized inertia matrix is the factor most influencing LCP solver failure. Experiment 3 additionally sought to determine a lower limit on the performance of LCP solvers on multi-rigid body contact problems: generalized inertia matrices may be poorly conditioned but should never truly be rank deficient.

Contact Jacobian matrices were produced as described in Experiment 1. Each reduced rank generalized inertia matrix was produced by summing $r < n$ randomly generated rank-1 updates to the $n \times n$ zero matrix.

F. Experiment 4: reduced rank generalized inertia matrices, reduced row rank contact Jacobians

This experiment tested the combined effect of reduced rank in both generalized inertia matrices and contact Jacobian matrices. Generalized inertia matrices were constructed as described in Experiment 3, and contact Jacobians were constructed as described in Experiment 2.

Copositive LCP Solution Frequency					
			Parameterization		
Ex.	Method		Worst	Average	Best
1	Lemke		77.09%	91.08%	100.0%
1	PATH		67.21%	67.71%	67.82%
2	Lemke		53.35%	82.14%	99.91%
2	PATH		65.18%	66.29%	66.57%
3	Lemke		34.42%	36.38%	37.54%
3	PATH		27.07%	27.17%	27.20%
4	Lemke		26.99%	37.61%	43.16%
4	PATH		31.15%	31.30%	31.45%
5	Lemke	$[1, 10^5]$	52.85%	80.98%	99.94%
5		$[1, 10^8]$	48.24%	75.45%	99.84%
5		$[1, 10^{10}]$	42.95%	70.88%	99.72%
5	PATH	$[1, 10^5]$	67.44%	67.66%	67.72%
5		$[1, 10^8]$	67.75%	68.02%	68.14%
5		$[1, 10^{10}]$	61.61%	62.20%	62.44%
6	Lemke	$[1, 10^5]$	82.47%	95.99%	100.0%
6		$[1, 10^8]$	73.84%	91.32%	99.97%
6		$[1, 10^{10}]$	66.03%	85.97%	99.87%
6	PATH	$[1, 10^5]$	67.62%	67.63%	67.64%
6		$[1, 10^8]$	67.79%	67.90%	67.93%
6		$[1, 10^{10}]$	63.38%	63.62%	63.70%

TABLE IV.2

SUMMARY OF SOLVER PERFORMANCE ACROSS SETS OF COPOSITIVE LCPS FOR THE EXPERIMENTS. BEST AND WORST COLUMNS GIVE SOLUTION FREQUENCY FOR THE PARAMETERIZATION WITH THE GREATEST AND LEAST SUCCESS, RESPECTIVELY. AVERAGE SOLUTION FREQUENCY IS OVER TOTAL RANGE OF PARAMETERS CONSIDERED.

G. Experiment 5: generalized inertia matrices with eigenvalues in intervals $[1, 10^5]$, $[1, 10^8]$, and $[1, 10^{10}]$, rank-deficient contact Jacobians

Tables IV.1 and IV.2, which summarize solubility on the previous experiments, show that the rank of the generalized inertia matrix is a much greater factor in LCP solubility than the row rank of the contact Jacobian matrices. As stated above, generalized inertia matrices should never truly be rank deficient. Accordingly, Experiment 5 gauges the effect of poorly conditioned inertia matrices on LCP solubility. We test solver performance on problems with generalized inertia matrices with eigenvalues ranging from $[1, 10^5]$, from $[1, 10^8]$, and from $[1, 10^{10}]$ (yielding condition numbers of 10^5, 10^8, and 10^{10}, respectively). Matrices with such condition numbers can arise due to large disparities in moments-of-inertia (which are dependent upon both the distribution and magnitude of a body's mass), even if there are not disparities of orders of magnitude between bodies' masses.

We generate inertia matrices with these condition numbers by starting with summation of n randomly generated rank-1 updates to an $n \times n$ zero matrix. A decomposition of this matrix yields the n-dimensional vector of eigenvalues $\boldsymbol{\lambda}$ and an $n \times n$ matrix $\mathbf{Q}$ with columns composed of linearly independent eigenvectors. We then exponentially scale the eigenvalues into the desired interval. We compose the inertia matrix through the operation[3] $\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{-1}$, where $\boldsymbol{\Lambda}$ is the diagonal matrix composed from the elements of $\boldsymbol{\lambda}$.

H. Experiment 6: effect of regularization

Experiment 6 constructs problems identically to Experiment 5 but employs regularization in the LCP solver as necessary to boost solubility. Thus, Experiment 6 indicates

[3]This operation is consistent with the spectral theorem for matrices.

the expected solubility and running time on practical multi-rigid body contact problems with state of the art solvers.

V. RESULTS AND OBSERVATIONS

Our experimental results are summarized in Tables IV.1–IV.6. The large volume of experimental results stymies attempts at visualization and hampers discernment of trends from tabulated data. To enhance data presentation, we provide running times and constraint violation for only Experiment 6 in Tables IV.3–IV.6. We rationalized that only Experiment 6 reflects both results with regularized algorithms and expected (*i.e.*, ill-conditioned but full rank generalized inertia matrices and reduced-rank contact Jacobians) data. Tables IV.1–IV.6 yield the following observations:

- Contact problems corresponding to monotone LCPs are "easier" to solve (evinced by higher success rate) than copositive LCPs. This result argues in favor of convex contact models, as advocated in [9], even though regularization mitigates much of this advantage (at cost of significant additional running time).
- Reducing the rank of the contact Jacobians lowers the solubility of the problem. If the rank of the generalized inertia matrix is reduced instead, the solubility of the problem is lowered much further. However, the solubility of the reduced rank for both is counterintuitively *higher* than the case of reduced rank generalized inertia and full rank contact Jacobian matrices (*i.e.*, solubility results for Experiment 4 are, in general, considerably higher than those for Experiment 3).
- With respect to solubility, LEMKE is the best solver. Our interior-point solver is not recommended for solubility or running time (at least on these problem sizes), though it is extremely robust with respect to parameter selection.
- With regard to all evaluation criteria (solubility, running time, and constraint violation), PATH performs better over a wider range of parameters than LEMKE.
- LEMKE's running time is significantly faster than that of PATH on the monotone LCPs, yet it runs significantly more slowly on the generated copositive LCPs.
- Constraint violation performance of PATH *vs*. LEMKE is comparable on average. In the worst cases (those reported in the "max" column under the best parameterizations in Tables IV.5 and IV.6), LEMKE performs considerably better than PATH on the monotone problems and considerably worse on the copositive problems.

Finally, we note that the ceilings on generalized coordinate dimension (15) and numbers of contact points (20)—selected to make the experiments tractably computable—do have (limited) effect on solubility. In an additional experiment that raised both ceilings to 100, applying LEMKE to copositive LCP contact problems with condition numbers distributed in $[1, 10^{10}]$ (*i.e.*, constructed in the manner of Experiment 5), maximum solubility reduced to 91.77% (from 99.72%).

VI. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the critiques from the anonymous reviewers. This work was funded by award CMMI-1100532 from the National Science Foundation.

REFERENCES

[1] P. Löstedt, "Mechanical systems of rigid bodies subject to unilateral constraints," *SIAM Journal on Applied Mathematics*, vol. 42, no. 2, pp. 281–296, 1982.

[2] J. J. Moreau, *Standard inelastic shocks and the dynamics of unilateral constraints*. New York: Springer-Verlag, 1983, pp. 173–221.

[3] D. Baraff, "Fast contact force computation for nonpenetrating rigid bodies," in *Proc. of SIGGRAPH*, Orlando, FL, July 1994.

[4] D. E. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction," *Intl. Journal for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2673–2691, 1996.

[5] D. Stewart and J. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with coulomb friction," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, San Francisco, CA, 2000.

[6] M. Anitescu and F. Potra, "Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems," *Nonlinear Dyn.*, vol. 14, pp. 231–247, 1997.

[7] M. Anitescu and F. A. Potra, "A time-stepping method for stiff multi-rigid-body dynamics with contact and friction," *Intl. Journal for Numerical Methods in Engineering*, vol. 55, pp. 753–784, 2002.

[8] E. Drumwright and D. A. Shell, "A robust and tractable contact model for dynamic robotic simulation," in *Proc. of ACM Symp. on Applied Computing (SAC)*, 2009, pp. 1176–1180.

[9] E. Drumwright and D. Shell, "Modeling contact friction and joint friction in dynamic robotic simualtion using the principle of maximum dissipation," in *Tracts in Advanced Robotics: Algorithmic Foundations of Robotics IX*. Springer, 2010, pp. 249–266.

[10] G. B. Dantzig and R. W. Cottle, "Positive (semi-)definite programming," in *Nonlinear Programming*, J. Abadie, Ed. Amsterdam: North Holland, 1967, pp. 55–73.

[11] R. W. Cottle, "The principal pivoting method of quadratic programming," in *Mathematics of Decision Sciences*, G. Dantzig and J. A. F. Veinott, Eds. Rhode Island: AMS, 1968, pp. 144–162.

[12] K. Murty, "Notes on a bard-type scheme for solving the complementarity problem," *Opsearch*, vol. 11, 1974.

[13] C. E. Lemke, "Bimatrix equilibrium points and mathematical programming," *Management Science*, vol. 11, pp. 681–689, 1965.

[14] R. W. Cottle, J.-S. Pang, and R. Stone, *The Linear Complementarity Problem*. Boston: Academic Press, 1992.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[16] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer-Verlag, 1999.

[17] S. P. Dirkse and M. Ferris, "The PATH solver: A non-monotone stabilization scheme for mixed complementarity problems," *Optimization Methods and Software*, vol. 5, pp. 123–156, 1995.

[18] G. D. Hart and M. Anitescu, "A hard-constraint time-stepping approach for rigid multibody dynamics with joints, contact, and friction," in *Proc. of Conf. on Diversity in Computer*, 2003, pp. 34–41.

[19] S. Berard, J. Trinkle, B. Nguyen, B. Roghani, V. Kumar, and J. Fink, "daVinci code: a multi-model simulation and analysis tool for multibody systems," in *Proc. of IEEE Conf. on Robotics and Automation (ICRA)*, April 2007, pp. 2588–2593.

[20] S. Berard, B. Nguyen, and J. Trinkle, "Sources of error in a rigid body simulation of rigid parts on a vibrating rigid plate," in *Proc. of ACM Symp. on Applied Computing*, Honolulu, Hawaii, May 2009, pp. 1181–1185.

[21] K. G. Murty, *Linear complementarity, linear and nonlinear programming*, ser. Sigma Series in Applied Mathematics 3. Berlin: Heldermann-Verlag, 1988.

[22] C. Lacoursière, "Splitting methods for dry frictional contact problems in rigid multibody systems: Preliminary performance results," in *Proc. of SIGRAD*, M. Ollila, Ed., Nov 2003, pp. 11–16.

[23] A. Tikhonov, "Solution of incorrectly formulated problems and the regularization method," *Soviet Mah. Dokl.*, vol. 4, pp. 1035–1038, 1963.

[24] P. Fackler and M. Miranda, "Implementation of a modified lemke's complementary pivoting algorithm," http://people.sc.fsu.edu/ burkardt/m_src/lemke/lemke.m, 2002.

Running times for solution of various Monotone LCPs						
			Best Parameterization		All Parameterizations	
Ex.	Method		Mean (σ)	Max	Mean (σ)	Max
6	Lemke	$[1, 10^5]$	1.40×10^{-4} (0.001)	0.03	3.93×10^{-4} (0.004)	0.64
			1.40×10^{-4} (0.001)	0.03	4.20×10^{-4} (0.004)	0.64
		$[1, 10^8]$	1.90×10^{-4} (0.001)	0.02	4.63×10^{-4} (0.006)	0.78
			1.90×10^{-4} (0.001)	0.02	5.28×10^{-4} (0.006)	0.78
		$[1, 10^{10}]$	1.88×10^{-4} (0.001)	0.01	4.05×10^{-4} (0.003)	0.34
			1.88×10^{-4} (0.001)	0.01	5.50×10^{-4} (0.004)	0.44
6	PATH	$[1, 10^5]$	8.67×10^{-4} (0.003)	0.05	1.92×10^{-3} (0.006)	0.10
			8.67×10^{-4} (0.003)	0.05	1.92×10^{-3} (0.006)	0.10
		$[1, 10^8]$	1.07×10^{-3} (0.003)	0.06	2.16×10^{-3} (0.007)	0.09
			1.08×10^{-3} (0.003)	0.06	2.38×10^{-3} (0.008)	0.12
		$[1, 10^{10}]$	1.76×10^{-3} (0.006)	0.11	2.37×10^{-3} (0.007)	0.12
			3.42×10^{-3} (0.014)	0.15	4.53×10^{-3} (0.016)	0.16
6	IP	$[1, 10^5]$	4.13×10^{-1} (1.185)	20.12	4.11×10^{-1} (1.186)	20.26
			1.09 (3.291)	25.66	1.09 (3.290)	25.84
		$[1, 10^8]$	3.84×10^{-1} (0.819)	13.93	3.70×10^{-1} (0.815)	14.29
			2.59 (5.067)	28.22	2.58 (5.068)	28.26
		$[1, 10^{10}]$	3.12×10^{-1} (0.735)	17.89	2.97×10^{-1} (0.731)	17.93
			3.13 (5.521)	27.21	3.12 (5.524)	27.27
Key: Plain Row=Timing for Solved LCPs / Shaded Row=Timing for All LCPs						

TABLE IV.3

SUMMARY OF SOLVER SPEED ON MONOTONE LCPS FOR EXPERIMENT 6. MEAN, STANDARD DEVIATION, AND MAXIMUM RUNNING TIMES ARE REPORTED FOR BOTH THE PARAMETERIZATION WHICH SOLVED THE GREATEST NUMBER OF PROBLEMS (LABELED BEST) AND OVER ALL PARAMETERIZATIONS. ROWS SHADING DISTINGUISH BETWEEN STATISTICS FOR PROBLEMS IN WHICH A SOLUTION WAS FOUND (WHITE), AND ALL PROBLEMS ATTEMPTED (SHADED).

Running times for solution of random copositive LCPs						
			Best Parameterization		All Parameterizations	
Ex.	Method		Mean (σ)	Max	Mean (σ)	Max
6	Lemke	$[1, 10^5]$	5.94×10^{-1} (2.018)	47.54	1.41 (15.212)	1781.61
			6.53×10^{-1} (2.209)	47.54	1.80 (15.961)	1781.61
		$[1, 10^8]$	6.67×10^{-1} (1.249)	7.09	1.61 (7.149)	262.66
			9.02×10^{-1} (2.541)	28.84	2.61 (9.620)	262.66
		$[1, 10^{10}]$	5.16×10^{-1} (1.167)	8.35	1.24 (7.266)	346.54
			1.02 (5.576)	111.53	2.48 (11.006)	346.54
6	PATH	$[1, 10^5]$	1.47×10^{-2} (0.043)	0.82	1.66×10^{-2} (0.059)	2.94
			3.65×10^{-2} (0.285)	6.08	3.92×10^{-2} (0.290)	6.23
		$[1, 10^8]$	5.29×10^{-2} (0.468)	7.78	5.07×10^{-2} (0.437)	11.35
			8.88×10^{-2} (0.570)	7.78	1.19×10^{-1} (0.918)	27.79
		$[1, 10^{10}]$	2.38×10^{-2} (0.173)	2.79	3.51×10^{-2} (0.423)	20.02
			2.21×10^{-1} (1.867)	34.32	3.34×10^{-1} (3.076)	81.01
Key: Plain Row=Timing for Solved LCPs / Shaded Row=Timing for All LCPs						

TABLE IV.4

SUMMARY OF SOLVER SPEED ON COPOSITIVE LCPS FOR EXPERIMENT 6. MEAN, STANDARD DEVIATION, AND MAXIMUM RUNNING TIMES ARE REPORTED FOR BOTH THE PARAMETERIZATION WHICH SOLVED THE GREATEST NUMBER OF PROBLEMS (LABELED BEST) AND OVER ALL PARAMETERIZATIONS. ROWS SHADING DISTINGUISH BETWEEN STATISTICS FOR PROBLEMS IN WHICH A SOLUTION WAS FOUND (WHITE), AND ALL PROBLEMS ATTEMPTED (SHADED).

Constraint violation (along the contact normal) for solved monotone LCPs							
			Best Parameterization			All Parameterizations	
Ex.	Method		Mean	(σ)	Max	Mean (σ)	Max
6	Lemke	$[1, 10^5]$	7.13×10^{-10}	(2.89×10^{-8})	1.72×10^{-6}	1.72×10^{-7} (1.01×10^{-5})	9.67×10^{-4}
		$[1, 10^8]$	9.75×10^{-9}	(3.74×10^{-7})	2.27×10^{-5}	5.33×10^{-7} (1.14×10^{-5})	9.82×10^{-4}
		$[1, 10^{10}]$	4.39×10^{-7}	(1.10×10^{-5})	6.70×10^{-4}	1.35×10^{-6} (2.26×10^{-5})	9.61×10^{-4}
6	PATH	$[1, 10^5]$	7.50×10^{-8}	(1.77×10^{-6})	1.12×10^{-4}	4.15×10^{-7} (1.09×10^{-5})	7.60×10^{-4}
		$[1, 10^8]$	6.06×10^{-8}	(2.66×10^{-7})	1.42×10^{-5}	5.84×10^{-7} (1.31×10^{-5})	7.44×10^{-4}
		$[1, 10^{10}]$	4.27×10^{-7}	(1.44×10^{-5})	6.70×10^{-4}	9.28×10^{-7} (2.09×10^{-5})	9.07×10^{-4}
6	IP	$[1, 10^5]$	3.57×10^{-7}	(8.83×10^{-6})	2.84×10^{-4}	3.57×10^{-7} (8.83×10^{-6})	2.84×10^{-4}
		$[1, 10^8]$	8.23×10^{-7}	(1.90×10^{-5})	9.47×10^{-4}	8.23×10^{-7} (1.90×10^{-5})	9.47×10^{-4}
		$[1, 10^{10}]$	1.91×10^{-5}	(8.22×10^{-5})	9.95×10^{-4}	1.91×10^{-5} (8.22×10^{-5})	9.95×10^{-4}

TABLE IV.5

SUMMARY OF MONOTONE LCP SOLUTION QUALITY FOR EXPERIMENT 6. MEAN, STANDARD DEVIATION, AND MAXIMUM VALUES OF THE NORMAL CONSTRAINT VIOLATION ARE REPORTED FOR BOTH THE PARAMETERIZATION WHICH SOLVED THE GREATEST NUMBER OF PROBLEMS AND OVER ALL PARAMETERIZATIONS.

Constraint violation (along the contact normal) for solved copositive LCPs							
			Best Parameterization			All Parameterizations	
Ex.	Method		Mean	(σ)	Max	Mean (σ)	Max
6	Lemke	$[1, 10^5]$	4.69×10^{-6}	(4.51×10^{-5})	8.58×10^{-4}	2.56×10^{-6} (3.01×10^{-5})	9.08×10^{-4}
		$[1, 10^8]$	4.72×10^{-6}	(5.68×10^{-5})	9.81×10^{-4}	2.77×10^{-6} (3.50×10^{-5})	9.81×10^{-4}
		$[1, 10^{10}]$	4.64×10^{-7}	(3.23×10^{-6})	4.85×10^{-5}	4.44×10^{-6} (4.91×10^{-5})	9.55×10^{-4}
6	PATH	$[1, 10^5]$	3.79×10^{-8}	(1.09×10^{-7})	9.93×10^{-7}	8.79×10^{-8} (4.52×10^{-6})	4.22×10^{-4}
		$[1, 10^8]$	5.84×10^{-8}	(1.35×10^{-7})	7.50×10^{-7}	1.66×10^{-7} (4.79×10^{-6})	2.58×10^{-4}
		$[1, 10^{10}]$	8.04×10^{-8}	(1.77×10^{-7})	1.23×10^{-6}	4.70×10^{-7} (1.59×10^{-5})	9.62×10^{-4}

TABLE IV.6

SUMMARY OF COPOSITIVE LCP SOLUTION QUALITY FOR EXPERIMENT 6. MEAN, STANDARD DEVIATION, AND MAXIMUM VALUES OF THE NORMAL CONSTRAINT VIOLATION ARE REPORTED FOR BOTH THE PARAMETERIZATION WHICH SOLVED THE GREATEST NUMBER OF PROBLEMS AND OVER ALL PARAMETERIZATIONS.